This listing of claims will replace all prior versions, and listings, of claims in the application:

The Status of the Claims

1. (Currently Amended) A method to optimize a program comprising:

cold translating a plurality of blocks of the program from a first language to a second language to generate a cold translated program;

determining a cold execution trip count associated with a first one of the blocks in the cold translated program;

identifying the first block for hot translation when the cold execution trip count exceeds a first threshold;

when the first block is identified for hot translation, hot translating the first block into a first hot translated block by inserting a hot execution trip counting instruction into the first block ~~instructions~~ to calculate a hot execution trip count for the first hot translated block ~~if the cold execution trip count is less than a predetermined trip count threshold~~;

linking the first hot translated block into the cold translated program;

executing the cold translated program with the first hot translated block;

identifying the first hot translated block as associated with a hot loop when the hot execution trip count exceeds a second threshold;

~~identifying a loop in the translated program that is a candidate for optimization using profile data;~~

inserting instrumentation into the hot loop to develop profile data for a load instruction within the instrumented hot loop;

linking the instrumented hot translated block into the cold translated program;

executing the cold translated program with the instrumented hot loop; and

inserting a prefetching instruction into the hot loop if the profile data indicates **[[**~~a~~**]]** the load instruction in the instrumented hot loop meets a predefined criteria, the prefetching instruction to prefetch data for the load instruction.

2. (Currently Amended) A method as defined in claim 1 wherein inserting instrumentation into the hot loop comprises:

finding the **[[**~~a~~**]]** load instruction in the hot loop; and

inserting a first instruction sequence to record addresses associated with the load instruction.

3. (Currently Amended) A method as defined in claim 2 wherein the first instruction sequence causes the addresses to be recorded in a buffer associated with the hot loop, and inserting instrumentation into the hot loop further comprises:

inserting a second instruction sequence into the hot loop to trigger processing of the addresses in the buffer to determine if the profile data indicates the **[[**~~a~~**]]** load instruction in the hot loop meets the **[[**~~a~~**]]** predefined criteria.

4. (Currently Amended) A method as defined in claim 1 wherein the profile data identifies the load instruction as at least one of a single stride load, a multiple stride load, a cross stride load, and a base load of a cross stride load.

5. (Currently Amended) A method to optimize a program comprising:

cold translating the program from a first instruction set to a second instruction set to generate a cold translated program;

executing the cold translated program;

hot translating at least one block of the cold translated program to generate at least one hot translated block;

linking the at least one hot translated block into the cold translated program to generate a hot translated program;

executing the hot translated program;

identifying a hot loop ~~in the translated program~~ associated with the hot translated block that meets a first predefined criteria;

gen-translating the hot loop to insert instrumentation into the hot loop to attempt to identify a load which would benefit from prefetching; and

if the hot loop identifies the load ~~meets a second predefined criteria~~, use-translating the hot loop to insert a prefetch instruction associated with the load.

6. (Currently Amended) A method as defined in claim 5 wherein cold translating the program comprises:

identifying a block in ~~a foreign~~ the program; and

~~inserting instructions~~ associating a first counter with ~~to update a first counter into~~ ~~an~~ the ~~instruction~~ block to determine the number of times the ~~instruction~~ block is executed; and

wherein the method further comprises analyzing the first counter to determine if the block is a candidate for optimization.

7. (Original) A method as defined in claim 5 wherein gen-translating and use-translating the program each comprises translating the first instruction set to an intermediate instruction set and translating the intermediate instruction set to the second instruction set.

8. (Original) A method as defined in claim 7 wherein the intermediate instruction set comprises an instruction set different than the first instruction set and different than the second instruction set.

9. (Currently Amended) A method as defined in claim 5 wherein identifying the hot loop ~~in the translated program~~ comprises conditioning the hot **[[**~~a~~**]]** loop by a least common specialization operation.

10. (Currently Amended) A method as defined in claim 9 wherein the least common specialization operation comprises:

identifying a block of instructions associated with the hot loop that is a least common denominator block with other loops; and

rotating the loop such that the least common denominator block is a head of the hot loop.

11. (Currently Amended) A method as defined in claim 5 wherein identifying the hot loop ~~in the translated program~~ comprises:

using at least one of a cold execution trip count to determine the average number of times the hot loop is executed ~~during cold execution~~ when executing the cold translated program or a hot execution trip count to determine the number of times the hot loop is executed when executing the hot translated program.

12. (Currently Amended) A method as defined in claim 11 wherein the cold trip count ~~comprises instructions to determine~~ is calculated from the frequency a loop entry block is taken and the frequency the loop back edge is taken.

13. (Currently Amended) A method as defined in claim 11 wherein gen-translating the hot loop is ~~gen-translated if~~ performed when the hot loop contains a load instruction and when a value of at least one of the **[[**~~a~~**]]** hot execution trip count and the **[[**~~a~~**]]** cold execution trip count is greater than a ~~predetermined~~ threshold.

14. (Currently Amended) A method as defined in claim 13 wherein gen-translating the hot loop is only ~~gen-translated~~ performed if the load instruction does not access data in a stack or have a loop invariant load address.

15. (Currently Amended) A method as defined in claim 13 wherein the hot loop is optimized ~~by a normal hot translation if~~ without gen-translating the hot loop when the cold execution trip count is less than the ~~predetermined~~ threshold.

16. (Currently Amended) A method as defined in claim 5 wherein gen-translating the hot loop comprises:

identifying a load instruction within the hot loop;

inserting a profiling instruction in association with the load instruction;

inserting a profiling control instruction in a loop entry block of the loop to control the number of times the load instruction is profiled;

executing the profiling instruction to profile the load instruction; and

executing the profiling control instruction to determining if the load has been profiled more than a ~~predetermined~~ threshold number of times.

17. (Original) A method as defined in claim 16 wherein the profiling instruction comprises an instruction to assign the load instruction a unique identification number and an instruction to collect profiling information.

18. (Original) A method as defined in claim 17 wherein the unique identification number is stored with a data address of the load instruction.

19. (Currently Amended) A method as defined in claim 16 wherein the profiling instruction is to collect ~~information comprises~~ stride information.

20. (Original) A method as defined in claim 16 wherein the profiling control instruction comprises a counter to determine how many times the load instruction has been profiled.

21. (Currently Amended) A method as defined in claim 5 wherein use-translating comprises:

analyzing ~~the~~ profile information collected by the instrumentation~~; and~~

~~inserting a prefetching instruction for the load instruction~~.

22. (Original) A method as defined in claim 21 further comprising eliminating redundant prefetched loads.

23. (Original) A method as defined in claim 21 wherein analyzing the profile information comprises determining if the load instruction is at least one of: a single stride load, a multiple stride load, a cross stride load; and a base load.

24. (Currently Amended) A method as defined in claim 5 further comprising linking the use-translated hot loop into the ~~native~~ the cold translated program.

25. (Currently Amended) An apparatus having a logic circuit, the apparatus to optimize a program comprising:

a cold translator to translate the program from a first instruction set to a second instruction set to generate a cold translated program;

a hot translation module to hot translate at least one block in the cold translated program;

a code linker to link the at least one hot translated block into the cold translated program to generate a hot translated program;

a hot loop identifier to identify a hot loop in the hot translated program and to determine if the hot loop should be gen-translated[[~~.~~]];

a gen-translator to instrument the hot loop with instructions to collect profile information; and a use-translator to ~~optimize an~~ insert a prefetch instruction associated with the hot loop ~~if~~ based on the profile information ~~determines that the hot loop should be optimized~~.

26. (Currently Amended) An apparatus as defined in claim 25 wherein the hot loop identifier identifies a loop as a hot loop by:

counting a number of times an instruction block associated with the loop is executed;

determining an average number of times the loop is executed; and

comparing the average number of times the loop is executed to a ~~predetermined~~ threshold.

27. (Currently Amended) An apparatus as defined in claim 25 wherein the hot loop identifier identifies the **[[**~~a~~**]]** hot loop in the hot translated program by conditioning a first loop by a least common specialization operation.

28. (Currently Amended) An apparatus as defined in claim 27 wherein the least common specialization operation comprises:

identifying a block of instructions that is a least common denominator block with the first loop and other loops;

~~rotating the loop such that~~ making the least common denominator block ~~is~~ a head of the first loop.

29. (Original) An apparatus as defined in claim 25 wherein the gen-translator and the use-translator each translates the program from the first instruction set to an

intermediate instruction set and from the intermediate instruction set to the second instruction set.

30. (Currently Amended) An apparatus as defined in claim 25 wherein the gen-translator comprises:

a load instruction identifier to identify a load instruction within the hot loop ~~and~~ having at least one ~~predetermined~~ characteristic;

a profiler to insert profiling instructions into the hot loop if the load instruction identifier identifies the [[~~a~~]] load instruction within the hot loop having the at least one ~~predetermined~~ characteristic.

31. (Original) An apparatus as defined in claim 30 wherein the profiler collects stride information for the load instruction.

32. (Currently Amended) An apparatus as defined in claim 25 wherein the use-translator comprises:

a profile analyzer to determine a load instruction type for ~~the~~ a load instruction associated with the hot loop based on the profile information ~~data~~;

an optimizer to insert the [[~~a~~]] prefetch instruction into the hot loop for the load instruction; and

wherein the [[~~a~~]] code linker is to couple the hot loop to the hot translated program after the prefetch instruction is inserted into the hot loop.

33. (Original) An apparatus as defined in claim 32 wherein the optimizer determines an address to be prefetched based on the load instruction type.

34. (Original) An apparatus as defined in claim 32 wherein the load instruction type comprises at least one of: a single stride load, a multiple stride load, a cross stride load, and a base load of a cross stride load.

35. (Cancelled)

36. (Currently Amended) A machine readable medium as defined in claim ~~35~~ 37 wherein the load instruction comprises at least one of: a single stride load, a multiple stride load, a cross stride load, and a base load of the cross stride load.

Please add the following new claim:

37. (New) A machine readable medium storing instructions structured to cause a machine to:

cold translate a program from a first language to a second language to generate a cold translated program;

determine a cold execution trip count associated with a first one of the blocks of the cold translated program;

identify the first block for hot translation when the cold execution trip count exceeds a first threshold;

when the first block is identified for hot translation, hot translate the first block into a first hot translated block by inserting a hot execution trip counting instruction into the first block to calculate a hot execution trip count for the first hot translated block;

link the hot translated block into the cold translated program to generate a hot translated program

execute the hot translated program;

identify the first hot translated block as associated with a hot loop when the hot execution trip count exceeds a second threshold;

insert instrumentation into the hot loop to develop profile data for a load instruction within the instrumented hot loop;

link the instrumented hot loop into the hot translated program;

execute the instrumented hot loop; and

insert a prefetching instruction into the hot loop if the profile data indicates the load instruction in the instrumented hot loop meets a criteria, the prefetching instruction to prefetch data for the load instruction.